Cinnober Financial Technology AB
Kungsgatan 36
SE-111 35 Stockholm
Sweden
+46 8 503 047 00
cinnober.com

TRADExpress™ RealTime Clearing FIX 4.4 Specification for APEX

Release version 1.7.0

Rev 1.0.2 Apr 19, 2018

This is the FIX specification for the APEX RTC system

Table of contents

About this document

TRADExpress™ RealTime Clearing implements a subset of the FIX 4.4 protocol.

Intended audience

This document is intended for clearing members and other parties that need to interact with the clearing system.

Related documents

FINANCIAL INFORMATION EXCHANGE PROTOCOL (FIX), Version 4.4

Revision history

This document has been revised according to the following table:

Revision state	Author	Comment
Rev 0.1.0 May 4, 2017	Anna Theorin	Initial version, functional description as per release 0.2.0
Rev 0.2.0 June 19,2017	Anna Theorin	Removed allowed values from Venue ID and instead state Enum 0-4 that is supported. Description is configurable as per Pending rule requirements.
Rev 0.3.0 Sept 11, 2017	Anna Theorin	Walkthrough of the document, correcting spelling mistakes and mistakes regarding references. Added symbol, NoUnderlyings, UnderlyingSymbol for Request for positions and Request for Trade.
Rev 0.4.0 Sept 29,2017	Anna Theorin	Updated/Changed description on fields 5179, 60, 75 on TCRs. Updated description of Business Level Party Identifiers Conventions. Added new enumerations (76=Desk ID and 36=Entering trader) in tag 452 (PartyRole). Added tags 55 (Symbol), 711 (NoUnderlyer) and 311 (UnderlyingSymbol) for Request for position subscription and Trade Capture Request subscription. Added tags 20009 (TradeEventDescription), 5442 (MatchingSlipID), and 943 (TimeBracket) to TCR. Corrected formatting in the sections Reject Handling and Authorization. Added General spelling and formatting corrections. Minor clarifications to tag descriptions.

Revision state	Author	Comment
Rev 0.5.0 Oct 30,2017	Tore Wennberg	Corrected valid combinations of PartyIDSource and PartyRole. Removed tag 109 (ClientID). Changed description of the following time fields: • TransacTime(60) • TradeTime(5179 • TradeDate(75)
Rev 0.6.0 Nov 03, 2017	Anna Theorin	Added new enumerations for 20009 (TradeEventDesciptions)
Rev 0.7.0 Dec 01, 2017	Tore Wennberg	Removed the following enumerations from tag 20009 (TradeEventDescription): • 73 - Added by Position Transfer • 74 - Removed by Position Transfer Updated description of TransactTime(60) for outgoing TCRs.
Rev 0.8.0 Jan 12, 2018	Tore Wennberg	Added sections describing Trade Request (6.1) and Trade Subscription (6.2). Clarified behavior in Failure between RTC and Member or ISV (10.2) Updated tag 5179 to state that it's a time (not date) field and UTC time not relevant.
Rev 0.9.0 Feb 22, 2018	Tore Wennberg	Corrected description of tag 555(NoLegs) in TCRs sent from FSMD and FSMEM.
Rev 1.0.0 Mar 13, 2018	Tore Wennberg	Changed description of tag 5179 (TradeTime) for all version of the TCR message. Renamed tag 5475 from PromptDate to ExpiryDate.
Rev 1.0.1 Mar 15, 2018	Tore Wennberg	Tag 5475 (ExpiryDate) is no longer a required tag in TCR messages.
Rev 1.0.2 Apr 19, 2018	Anke Eliasson	Corrected broken links

Terms and acronyms

Term/Acronym	Description
CH	Clearing house
CM	Clearing member
FS	FIX Server
FSMD	FIX Server Market Data
FSMEM	FIX Server Member
MtM	Mark-to-market
RTC	TRADExpress™ RealTime Clearing
TCR	Trade capture report

1 Introduction

The document describes:

- FIX Session layer: A section describing the FIX session layer and supported session level messages.
- Business Layer Introduction: A section describing the user interaction and the main workflows.
- FIX server setup: A section describing the different FIX servers.
- Trade Capture: A section describing different ways to capture trades for further processing in the clearing system. Subsections include:
    - Reporting of matched trades
    - Trade confirmation
    - Bust trade
- Trade Management: A section describing management of matched trades. Subsections include:
    - Trade Subscription
    - Trade History
- Position Management: A section describing position management. Subsections include:
    - Update Position Exercise Quantity
    - Position status
- Market Data: A section describing the market data features.
- General Messages: A section describing general application messages.

Common Component Blocks: A list of the common blocks of information used in messages.

1.1 Provided Services

The RTC system accepts market data and trades from the exchange and distributes trades, positions and risk margins to the appropriate parties. The system provides functionality to request market data and pulling historic trades from the system.

The system also allows for trade busts and custom handling of option exercise.

2 FIX Session Layer

2.1 FIX Session Establishment

A FIX session is established by sending a Logon message. The FIX session is established between two parties, the sender and the target represented by the following tags in the Standard Message Header:

SenderCompID (tag 49). The party initiating the session.

TargetCompID (tag 56). The acceptor of the session as per configuration.

The FIX session is always initiated by the FIX client and accepted by the FIX server.

2.2 Start of Day/End of Day Procedures

A FIX Session can last forever, or until:

- A login message that specifies that the sequence numbers should be reset, using ResetSeqNumFlag=Y (tag 141).
- A SequenceReset message is sent by either side of the FIX session.

As defined by the market place.

2.3 Reject Handling

This section describes FIX reject handling.

The Reject message is used when a message is received but cannot be properly processed due to a session level rule violation. Some examples[1]:

- A message lacking a mandatory tag.
- A message with an incorrect value for a specific tag.
- A tag without a value.
- Unknown message type.
- A tag appears more than once.

If an application-level message passes the syntax checking on the FIX Session level it should then be processed at the business level. If this process detects an error condition a business level reject should be issued. Many business level messages have specific “reject” messages, which should be used[2]. Business level messages lacking specific “reject” messages should use the Business Message Reject message.

The exceptions to use the specific business level reject message are:

[1] All supported FIX Session level reject reasons are presented in section 2.5.5

[2] Volume 1 of the FIX 4.4 Specification depicts all cases were Business Message Reject should not be used, and most FIX messages actually have their own messages for application level reject, for example a Trade Capture Report is rejected via a Trade Capture Report Ack message. Allocation Instruction Ack on the other hand does not have any specific reject message so this message must be rejected via Business Message Reject. The Business Message Type section of each business section in this document describes which messages that are used for business level reject.

- In the event a business message is received, fulfils session-level rules, however, the message cannot be communicated to the business-level processing system. In this situation a Business Message Reject with BusinessRejectReason = “Application not available at this time” can be issued.
- In the event a valid business message is received, fulfils session-level rules, however, the message type is not supported by the recipient. In this situation a Business Message Reject message with BusinessRejectReason=”Unsupported Message Type” can be issued.
- In the event a business message is received, fulfils session level-rules, but lacks a field conditionally required by the FIX specification. However, a Business Message Reject message is NOT used to enforce proprietary rules more restrictive than those explicit in the FIX specification, such as requiring a Trade Capture Report to contain a TradeReportTransType, which the FIX specification considers to be optional.

2.4 Authorization

All FIX Sessions are subject to authorization. When the FIX gateway receives a Logon message at connection start up, the session is authorized using:

- SenderCompId (49). This is the session identifier together with the TargetCompId.
- Username (553). Must contain the member ID assigned by the Clearing House.
- Password (554). Must contain the password assigned by the Clearing House

2.5 Message Details

The following sections cover all supported Session Messages.

FIX Message Name	Type	Direction
Logon	A	In/Out
Heartbeat	0	In/Out
Test Request	1	In/Out
Resend Request	2	In/Out
Reject	3	Out
Sequence Request	4	In/Out
Logout	5	In/Out

2.5.1 Logon

The logon message authenticates a user establishing a connection to a remote system. The logon message must be the first message sent by the session initiator.

The FIX server requires an encrypted password. To encrypt the password, additional information is passed using the RawData <96> and RawDataLength <95> fields.

A varying number is used to further scramble the password and to make the encrypted password different for each logon. This number:

1. Is an integer number.
2. Has a value of the current (GMT) system time in milliseconds.
3. Is higher than the previous midnight's (GMT) time in milliseconds.
4. Is lower than the following midnight's (GMT) time in milliseconds.
5. Has a higher value than the number used in the previous logon request.

A letter identifies subfields. A colon delimits a subfield identifier from a subfield value. The delimiter for the client random number is 'm'. A typical RawData value would be m:1065641126118 from a client logging on to the first version of the FIX server.

Password Encryption

The following entities make up the encrypted password:

1. A hash pattern applied on the source password.
2. The 32 byte fax key, a field from the user's RTC account that is copied outside the protocol, probably by fax or email.
3. The "HmacSHA1" encryption algorithm.
4. An 8 byte increasing number generated by the client.

The Clearing House may assist third-party developers in finding the required source code in Java or C/C++ that implements the "HmacSHA1" encryption algorithm.

If the logon request is rejected, a logout message is sent back with the reason for the rejection in the Text <58> field and the TCP/IP session is terminated by the FIX Gateway.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = A
98	EncryptMethod	Y	Method of encryption. 0 = None.
108	HeartBtInt	Y	Heartbeat interval (seconds). Must be greater than zero.
95	RawDataLength	N	Number of bytes in raw data field.
96	RawData	N	Unformatted raw data, can include bitmaps, word processor documents, etc.
141	ResetSeqNumFlag	N	Indicates that both sides of the FIX session should reset sequence numbers.
789	NextExpectedMsgSeqNum	N	Optional, alternative via counterpart bi-lateral agreement message gap detection and recovery approach. (for future use)
553	Username	Y	Must contain the member ID assigned by the Clearing House
554	Password	Y	Must contain the password assigned by the Clearing House

Tag	Field Name	Req'd	Comments
	Standard Trailer	Y	

2.5.2 Heartbeat

During periods of message inactivity, FIX applications will generate Heartbeat messages at regular time intervals. The heartbeat monitors the status of the communication link and identifies incoming sequence number gaps.

When logging on, the client requests a heartbeat interval, using the HeartBtInt tag (see the logon message). Heartbeats must be sent in both directions:

- The FIX Gateway will send Heartbeat requests at the requested interval, unless other messages are sent.
- The FIX Client must send Heartbeat requests at the requested interval, unless other messages are sent.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 0
112	TestReqID	N	Required when the heartbeat is a result of a Test Request message.
	Standard Trailer	Y	

2.5.3 TestRequest

The test request message forces a heartbeat from the opposing application. Test requests are responded to with a Heartbeat message conveying the test request ID.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 1
112	TestReqID	Y	The value will be returned in the resulting Heartbeat.
	Standard Trailer	Y	

2.5.4 ResendRequest

The resend request is sent to initiate the retransmission of messages.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 2
7	BeginSeqNo	Y	Sequence number of the first message in the range to be re-sent.
16	EndSeqNo	Y	Sequence number of the last message in the range to be re-sent. If the request is for all messages, subsequent to a particular message the *EndSeqNo* field should be set to 0 (representing infinity).
	Standard Trailer	Y	

2.5.5 Reject

See section 2.3 for usage.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 3
45	RefSeqNum	Y	MsgSeqNum of rejected message
371	RefTagID	N	The tag number of the FIX field being referenced.
372	RefMsgType	N	The MsgType of the FIX message being referenced.
373	SessionRejectReason	N	Code to identify the reason for a session-level Reject message: 0 = Invalid tag number. 1 = Required tag missing. 2 = Tag not defined for this message type. 3 = Undefined tag. 4 = Tag specified without a value. 5 = Value is incorrect (out of range) for this tag. 6 = Incorrect data format for value. 7 = Decryption problem. 9 = CompID problem. 10 = *SendingTime* accuracy problem. 11 = Invalid *MsgType*. 13 = Tag appears more than once. 14 = Tag specified out of required order. 15 = Repeating group fields out of order. 16 = Incorrect NumInGroup count for repeating group. 17 = Non "data" value includes field delimiter (SOH character). 98 = Service not available at this time. 99 = Other.
58	Text	N	Where possible, message to explain reason for rejection
	Standard Trailer	Y	

2.5.6 Sequence Reset

A sequence reset has two modes: Gap Fill Mode which will be used as the response to a Resend request and Reset Mode used to reset the sequence number after an unrecoverable application failure. A sequence reset – Reset Mode can only increase the sequence number.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 4
36	NewSeqNo	Y	New sequence number.

Tag	Field Name	Req'd	Comments

123	GapFillFlag	N	Indicates that the Sequence Reset message is replacing administrative or application messages which will not be resent.
			Y = Gap Fill message, *MsgSeqNum* field valid.
			N = Sequence Reset, ignore *MsgSeqNum*.

	Standard Trailer	Y	

2.5.7 Logout

The logout message initiates or confirms the termination of a FIX session. FIX clients should terminate their sessions gracefully by logging out.

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = 5
58	Text	N	Additional description of the message.
	Standard Trailer	Y	

3 Business Layer Introduction

3.1 User Interaction

Trades may come from an external trading system, or a clearing operator depending on system set up. Trades and position updates are sent unsolicited to the users. The users will get unsolicited trades and positions for all trades and positions they are authorized for and subscribed to.

It is also possible to perform a snapshot request for historic trades and current positions.

4 FIX servers

4.1 Overview

The system contains four different types of FIX servers:

- FS: Incoming flow of matched trades from the exchange as well as busts.
- FSMD: Query and subscription service for trades, positions, market data and news. Also, supports requests for historic trades and position maintenance reports.
- FSMEM: Server for outgoing trade capture reports with support for session layer resend requests.
- FCMD: Connects to external market data service for incoming market data.

4.2 Supported Messages

Below is a list of supported FIX messages per FIX server and direction.

Message	MsgType	FS		FSMD		FSMEM		FCMD	
		In	Out	In	Out	In	Out	In	Out
Logon	A	X	X	X	X	X	X	X	X
Heartbeat	0	X	X	X	X	X	X	X	X
TestRequest	1	X	X	X	X	X	X	X	X
ResendRequest	2	X	X			X	X		
Reject	3		X		X		X	X	
SequenceReset	4	X	X	X	X	X	X	X	X
Logout	5	X	X	X	X	X	X	X	X
BusinessMessageReject	j	X	X		X		X	X	
TradeCaptureReportRequest	AD			X					
TradeCaptureReportRequestAck	AQ				X				
TradeCaptureReport	AE	X	X		X		X		
TradeCaptureReportAck	AR		X						
RequestForPositions	AN			X					
RequestForPositionsAck	AO				X				
PositionReport	AP				X				
PositionMaintenanceRequest	AL			X					
PositionMaintenanceReport	AM				X				
MarketDataRequest	V			X					X
MarketDataSnapshotFullRefresh	W				X				
MarketDataRequestReject	Y				X			X	
MarketDataIncrementalRefresh	X							X	
News	B		X		X		X		
ApplicationMessageRequest	BW		X						

5 Trade Capture and Busts

This section describes the process of publishing trades and busts to the FS server and the TCR's published back to the exchange and users connected to FSMEM.

TCR's published by FSMD to subscribers are described in a later section.

5.1 Business Message Types

The FS server supports the message types described in the following table.

Fix Message Name	Type	Dir.	Comment
Trade Capture Report	AE	In	Used for the following purposes: ▪ Report matched trades from a trading venue. ▪ Update matched trade ▪ Bust matched trades
Trade Capture Report Ack	AR	Out	Used to acknowledge or reject an incoming Trade Capture Report
Trade Capture Report	AE	Out	Used for the following purposes: ▪ Send notification to counterpart(s) that does not require confirmation.

5.2 Trade Reporting Features

5.2.1 Trade Report Identification

Each Trade Capture Report message must be identified with a unique identification number, TradeReportID (tag 571).

In the message chaining model used for Trade Capture Reports a subsequent message may refer to a prior message via TradeReportRefID (tag 572).

The following rules apply to TCR identifiers:

- **TradeReportID** is assigned by the submitter of the message and used as a pure message identifier. This ID is always returned in the Trade Capture Report Ack.
- **TradeReportRefID** is assigned by the submitter when it wants to link a new message to a previous message in the same process.
    - The initiator always chains messages so that TradeReportRefID refers to the previous TradeReportID.

The clearing system uses the reference in confirmation messages when ending the process of reporting and acknowledging a privately negotiated trade and then refers to external actors via TradeReportID.

5.2.2 Trade Report Trans Type

TradeReportTransType, tag 487, designates the trade report transaction type. Used as action for unmatched trades.

- 0 – New
- 1 – Cancel

5.2.3 Trade Report Type

Tag 856, TradeReportType designates the type of the trade report. Used as action for matched trades.

- **0 - Submit**. This type is used when the counterpart response is not required.
- **6 - Trade Report Cancel**. This type is used to cancel/bust a matched trade or to indicate a cancelation of a trade as a result of giving up the trade.

5.2.4 Trade Identifiers

Trades have a number of identifiers used for different purposes:

- **ExecID, tag 17.**
  An external execution ID assigned to trades matched outside RTC. Same ID for buy and sell side.
- **TrdMatchID, tag 880.**
  A match identifier assigned by RTC to all matched trades. Same ID for buy and sell side. This is the identifier used when performing a trade bust or trade amendment of a matched trade.
- **SecondaryExecID, tag 527.**
  A match identifier assigned by RTC to each side of all matched trades. Different ID for buy and sell side. Note that SecondaryExecID is placed in the Side group.

5.2.5 Business Level Party Identifiers Conventions

All Inbound messages are subject to authorization and must therefore specify the party responsible for the business content of the message. The five supported types of party identifiers are Client ID, Client LEI, Clearing Member, Business Unit, and Trading User (a.k.a. Seat ID).

Each party identifier is represented by the three tags PartyID, PartyIDSource, and PartyRole:

- **PartyID (448)** = the specific party identifier.
- **PartyIDSource (447)**. Identifies class or source of the PartyID (448) value. Use D (Proprietary/Custom code) when supplying:

  Clearing Member
  Business Unit
  Trading User

  Use N (Legal Entity Identifier) when supplying:

  Client ID
  Client LEI
- **PartyRole (452)**. This is the party role for the specified party, and specifies the type of identifier that is populated in PartyID (448). Valid values are:

  3 – Client ID
  4 - Clearing Member (FIX Name: Clearing Firm)
  24 - Client LEI (FIX Name: Customer Account)
  36 - Trading User (FIX Name: Entering Trader)
  76 - Business Unit (FIX Name: Desk ID)

  Valid combinations of PartyIDSource and PartyRole are:

  PartyIDSource=D and PartyRole=3

PartyIDSource=D and PartyRole=36
PartyIDSource=D and PartyRole=76
PartyIDSource=N and PartyRole=4
PartyIDSource=N and PartyRole=24

5.2.6 ExecType

Tag 150, ExecType is only set on outgoing trade reports and trade capture report acknowledgements. The following values are:

- **0 - New**
  Indicates that RTC has accepted insert request.
- **8 - Rejected**
  Indicates that the trade report request was rejected.
- **9 – Suspended**

  Indicates that RTC has accepted the trade, but that it has not quite entered the system (i.e. on the Exception Queue). This is communicated as Paired or Accepted depending on the Trading Venue.
- **F - Trade**
  A new trade has entered RTC and it has been fully accepted into the system and is put on the requested Position Account. This is communicated as Accepted regardless of Trading Venue.
- **H - Trade Cancel**
  Indicates successful cancellation of a matched trade in RTC.

5.2.7 Text

The Text field is used to assign an optional trade reference. This trade reference can be set when reporting a trade and also modified when correcting a matched trade.

5.2.8 Instrument Legs Group (tag 10010)

The instrument legs group, together with Symbol(55) and CFICode(461), describe commodity instruments. Note that the InstrumentLegNo(20004) corresponds to LegInstrument(20005) on the NoLegs(555) group.

5.2.9 Legs Group (tag 555)

The legs group describes both sides of all the leg trades of the trade. LegInstrument(20005) links to InstrumentLegNo(20004) on the instrument leg group to identify the instrument. Leg Buy/Sell(20006) indicates if the individual leg is a buy or sell and LegSide(624) links to the Side group(552).

5.2.10 Trade Report Identifiers usage for Matched Trades Workflows

This table shows Trade Report Identifiers usage for matched trade workflows; that is, when reporting matched trades that do not require counterpart confirmation.

Action	487[3]	856[4]	571[5]	572[6]	880[7]
Initiators request (from initiator to marketplace)					
Insert matched trade	New	Submit	New client id	-	-
Cancel matched trade (bust)	New	Trade Report Cancel	New client id	-	RTC Trade id
Marketplace publication of confirmed trade					
Publication to initiator	New	Submit	New RTC id	Initiators id	RTC Trade id
Publication to counterparts	New	Submit	New RTC id		RTC Trade id

5.3 Message Details

5.3.1 Trade Capture Report (Incoming via FS)

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AE
1180	ApplID	N	Identifies the application with which a message is associated. Used only if application sequencing is in effect.
1181	ApplSeqNum	N	Data sequence number to be used when FIX session is not in effect
1350	ApplLastSeqNum	N	Application sequence number of last message in transmission
571	TradeReportID	Y	Unique identifier for the Trade Capture Report.
487	TradeReportTransType	Y	Identifies Trade Report message transaction type. Valid values: 0 = New
856	TradeReportType	Y	Type of Trade Report. RTC Valid values: 0 = Submit 6 = Trade Report Cancel
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as a cancelation.
880	TrdMatchID	N	A match ID assigned by RTC to all matched trades. Same ID for buy and sell side. This is the trade identifier and must be used to identify a trade when performing a bust.
17	ExecID	Y	Exchanged assigned Execution ID (Trade Identifier)

[3] 487 = TradeReportTransType

[4] 856 = TradeReportType

[5] 571 = TradeReportID

[6] 572 = TradeReportRefID

[7] 880 = TrdMatchID

Tag	Field Name		Req'd	Comments
570	PreviouslyReported		Y	Indicates if the trade capture report was previously reported to the counterpart In RTC always set to N = not reported to counterpart
5940	TradeNumber		N	Where a trade is reported via Trading system
7931	VenueID		N	Venue of execution. Enumeration with value 0-4 supported.
1301	MarketID		Y	Product Group for the trade. Allowed values are: • PG1 • PG2
	Instrument		Y	
31	LastPx		N	Option premium
943	TimeBracket		N	A code that represents a time interval in which a fill or trade occurred. A TCR must contain either a TimeBracket or a TradeTime (5179) per side.
75	TradeDate		Y	The system regards this as the logical business date for when the matched trade was submitted for clearing. Expressed in YYYYMMDD format.
60	TransactTime		Y	The time when the trade was executed/created at the Exchange. UTC Date Time field. Format: YYYYMMDD-HH:MM:SS.sss
555	NoLegs		Y	Specifies the number of Instrument Legs (repeating groups) for a trade. The number represents legs for both sides so is a multiple of 2.
→	624	LegSide	Y	1 or 2 to correspond to Side (54).
→	20005	LegInstrument	Y	Corresponds to InstrumentLegNo.
→	20006	LegBuySell	Y	• B - Buy • S - Sell
→	5474	AbbreviatedPrice	N	Contract Price. Price of the leg can be expressed as: (b) A price code expressions (e.g. S + 10 which means settlement price plus ten) Valid price codes are B(Basis), S, C, M (mean), MC and M3 is also supported as tags and will be mapped to M if received. This is only valid for futures (c) A differential (e.g. -10 which means ten units lower than the price of the first leg) A valid differential is a number prefixed with either (+) or (-). Note, current installation does not handle Abbreviated prices for allowed instruments.
→	198	SecondaryOrderID	N	Can be used to provide order ID used by exchange or executing system.

Tag	Field Name		Req'd	Comments
				In RTC, the unique order ID provided by the trading system.
→	10003	LegLastQty	N	Details the trade quantity for this leg. This tag is introduced as ID 1418 from 5.0SP1.
→	637	LegLastPx	N	Represents the price for the current trade leg.
→	5442	MatchingSlipID	Y	Identifier for middleware services that combine trade sides to full trades.
→	20007	UniqueTransaction ID	N	Unique Transaction Identifier, unique at half trade leg level. Note, not used by RTC
552	NoSides		Y	Number of sides. The number of trade halves in the structure.
→	54	Side	Y	In RTC valid values: 1 = First side 2 = Second side
→	37	OrderID	Y	OrderID is required to be unique for each chain of orders. In RTC, the unique client order id. Use “[N/A]” if not applicable.
→	11	ClOrdID	N	Client supplied order ID
→		**Parties**	N	For RTC, see section 12.4
→	1	Account	Y	Account
→	581	AccountType	Y	Type of account associated with the trade. Valid Values: 1 = ISA 2 = House 3 = Omnibus 9 = Unallocated House Account (U-Account) 90 = Unallocated Client Account (X-Account)
→	5681	ExchangeTradeType	N	Exchange defined type of trade. Allowed values: • 0 - NORMAL • 1 - BLOCK TRADE • 2 - EFRP • 3 - NIGHT TRADE
→	5476	PrivateReference	N	Private Reference up to 80 characters.
→	5477	PublicReference	N	Public Reference up to 12 characters.
→	5179	TradeTime	N	Time field representing the local time at the marketplace. Allowed formats: • HH:MM:SS • HH:MM:SS.sss

Tag	Field Name		Req'd	Comments
				Note, the system validates the time format, but will not perform any validation on the time it represents. A TCR must contain either a TimeBracket(943) or a TradeTime(5179) per side.
→	58	Text	N	In RTC used to specify an optional reference on a trade.
	Standard Trailer		Y	

Trade Capture Report Validation

There is validation on the TCR FIX message before it is accepted into the system. This includes, but is not limited to:

- The FIX syntax.
- That required tags are set.
- That tags values are within allowed limits.
- That the MarketID(1301) is valid for the Symbol(55).
- That the AccountType(581) is valid for the Account(1).
- That tags with dependencies fulfil these dependencies.

If any validation fails the trade will not be accepted into the system.

The response from RTC will be either of:

- Reject (MsgType=3)
- TradeCaptureReportAck (MsgType=AR) with ExecType(150) = 8 (Rejected).
- BusinessMessageReject (MsgType=j)

5.3.2 Trade Capture Report Ack

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AR
571	TradeReportID	Y	Unique identifier for the Trade Capture Report.
487	TradeReportTransType	N	Identifies Trade Report message transaction type. Same as in incoming Trade Capture Report.
856	TradeReportType	Y	Type of trade report. Same as in incoming Trade Capture Report.
939	TrdRptStatus	N	Status of Trade Report. Valid values: 0 = Accepted 1 = Rejected
150	ExecType	Y	Type of Execution being reported. Valid values: 0 = New 8 = Rejected See section 5.2.6 for more information.

Tag	Field Name	Req'd	Comments
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as correction or cancelation.
751	TradeReportRejectReason	N	0 = Successful (default) 1 = Invalid party information 2 = Unknown instrument 3 = Unauthorized to report trades 4 = Invalid trade type 99 = Other
527	SecondaryExecID	N	This is the trade ID assigned by RTC. Set when a new trade is accepted. Note: In a TCR message (AE) this trade ID is populated in the TrdMatchID (880).
1301	MarketID	Y	Product Group for the trade. Allowed values are: • PG1 • PG2
	Instrument	Y	
58	Text	N	Free format text string.
	Standard Trailer	Y	

5.3.3 Trade Capture Report (Outgoing from RTC via FS)

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AE
1180	ApplID	N	Identifies the application with which a message is associated. Used only if application sequencing is in effect.
1181	ApplSeqNum	N	Data sequence number to be used when FIX session is not in effect
1350	ApplLastSeqNum	N	Application sequence number of last message in transmission
571	TradeReportID	Y	Unique identifier for the Trade Capture Report.
487	TradeReportTransType	Y	Identifies Trade Report message transaction type. Valid values: 0 = New
856	TradeReportType	Y	Type of Trade Report. RTC Valid values: 0 = Submit 6 = Trade Report Cancel
855	SecondaryTrdType	N	Clearing trade event description.
150	ExecType	Y	Type of Execution being reported In RTC valid values: 8 = Rejected 9 = Suspended F = Trade H = Trade Cancel See 5.2.6 for more information.
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as a cancelation.

Tag	Field Name	Req'd	Comments
880	TrdMatchID	N	A match ID assigned by RTC to all matched trades. Same ID for buy and sell side. This is the trade identifier and must be used to identify a trade when performing a bust.
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier)
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterpart In RTC always set to N = not reported to counterpart
5940	TradeNumber	N	Where a trade is reported via Trading system, the trade reference will be included here.
7931	VenueID	N	Venue of execution. Enumeration with value 0-4 supported.
20009	TradeEventDescription	Y	Together with TradeReportType (tag 856) this field details the specific type of event. RTC valid values: 17 – New Trade Report 22 – Taken Up 23 – Given Up 56 – Busted 10009 – Trade Transfer In 10010 – Trade Transfer Out 10011 – Removed by Trade Split 10012 – Added by Trade Split 10013 – Partial Trade Transfer Out 10014 – Partial Trade Transfer In 10015 – Partial Give Up
1301	MarketID	Y	Product Group for the trade.
	Instrument	Y	
31	LastPx	N	Option premium, required if CFICode(461) defines an option contract. .
943	TimeBracket	N	A code that represents a time interval in which a fill or trade occurred.
75	TradeDate	Y	The system regards this as the logical business date for when the matched trade was submitted for clearing. Expressed in YYYYMMDD format.
60	TransactTime	Y	The UTC time when the trade was executed/created at the Exchange. If the TCR is sent regarding a trade bust, i.e. TradeReportType(856)=6, then TransactTime(60) will contain the UTC time when the trade was cancelled by RTC. UTC Date Time field. Format: YYYYMMDD-HH:MM:SS.sss
555	NoLegs	Y	Specifies the number of Instrument Legs (repeating groups) for a trade. The number

Tag	Field Name		Req'd	Comments
				represents legs for both sides so is a multiple of 2.
→	624	LegSide	Y	1 or 2 to correspond to Side (54).
→	20005	LegInstrument	Y	Corresponds to InstrumentLegNo.
→	20006	LegBuySell	Y	• B – Buy • S – Sell
→	5474	AbbreviatedPrice	N	Contract Price. Price of the leg can be expressed as: (b) A price code expressions (e.g. S + 10 which means settlement price plus ten) Valid price codes are B(Basis), S, C, M (mean), MC and M3 is also supported as tags and will be mapped to M if received. This is only valid for futures (c) A differential (e.g. -10 which means ten units lower than the price of the first leg) A valid differential is a number prefixed with either (+) or (-). Note, current installation does not handle Abbreviated prices for allowed instruments.
→	198	SecondaryOrderID	N	Can be used to provide order ID used by exchange or executing system. In RTC, the unique order ID provided by the trading system.
→	10003	LegLastQty	N	Details the trade quantity for this leg. This tag is introduced as ID 1418 from 5.0SP1.
→	637	LegLastPx	N	Represents the price for the current trade leg.
→	5442	MatchingSlipID	N	Identifier for middleware services that combine trade sides to full trades.
→	20007	UniqueTransaction ID	N	Unique Transaction Identifier, unique at half trade leg level. Note, not used by RTC
552	NoSides		Y	Number of sides. The number of trade halves in the structure.
→	54	Side	Y	In RTC valid values: 1 = First side 2 = Second side
→	527	SecondaryExecID	N	A match ID assigned by RTC to each side of all matched trades. Different ID for buy and sell side.
→	37	OrderID	Y	OrderID is required to be unique for each chain of orders. In RTC, the unique client order id. Use “[N/A]” if not applicable.
→	11	ClOrdID	N	Client supplied order ID
→		**Parties**	N	For RTC, see section 12.4
→	1	Account	Y	Account

Tag	Field Name		Req'd	Comments
→	581	AccountType	Y	Type of account associated with the trade. Valid Values: 1 = ISA 2 = House 3 = Omnibus 9 = Unallocated House Account (U-Account) 90 = Unallocated Client Account (X-Account)
→	5681	ExchangeTradeType	N	Exchange defined type of trade • 0 – NORMAL • 1 - BLOCK TRADE • 2 - EFRP • 3 - NIGHT TRADE
→	5476	PrivateReference	N	Private Reference up to 80 characters.
→	5477	PublicReference	N	Public Reference up to 12 characters.
→	5179	TradeTime	N	Time field representing the local time at the marketplace. Formats: • HH:MM:SS • HH:MM:SS.sss The value is copied from the TCR that was submitted to the FS.
→	58	Text	N	In RTC used to specify an optional reference on a trade.
	Standard Trailer		Y	

5.3.4 Trade Capture Report (Outgoing from RTC via FSMEM)

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AE
1180	ApplID	N	Identifies the application with which a message is associated. Used only if application sequencing is in effect.
1181	ApplSeqNum	N	Data sequence number to be used when FIX session is not in effect
1350	ApplLastSeqNum	N	Application sequence number of last message in transmission
571	TradeReportID	Y	Unique identifier for the Trade Capture Report.
487	TradeReportTransType	Y	Identifies Trade Report message transaction type. Valid values: 0 = New
856	TradeReportType	Y	Type of Trade Report. RTC Valid values: 0 = Submit 6 = Trade Report Cancel
855	SecondaryTrdType	N	Clearing trade event description.
150	ExecType	Y	Type of Execution being reported

Tag	Field Name	Req'd	Comments
			In RTC valid values: 8 = Rejected 9 = Suspended F = Trade H = Trade Cancel See 5.2.6 for more information.
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as a cancelation.
880	TrdMatchID	N	A match ID assigned by RTC to all matched trades. Same ID for buy and sell side. This is the trade identifier and must be used to identify a trade when performing a bust.
17	ExecID	N	Exchanged assigned Execution ID (Trade Identifier)
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterpart In RTC always set to N = not reported to counterpart
5940	TradeNumber	N	Where a trade is reported via Trading system the trade reference will be included here. An example would be TN-AA-20100331-00001.
7931	VenueID	N	Venue of execution. Enumeration with value 0-4 supported.
20009	TradeEventDescription	Y	Together with TradeReportType (tag 856) this field details the specific type of event. RTC valid values: 17 – New Trade Report 22 – Taken Up 23 – Given Up 56 – Busted 10009 – Trade Transfer In 10010 – Trade Transfer Out 10011 – Removed by Trade Split 10012 – Added by Trade Split 10013 – Partial Trade Transfer Out 10014 – Partial Trade Transfer In 10015 – Partial Give Up
	Instrument	Y	
31	LastPx	N	Option premium.
943	TimeBracket	N	A code that represents a time interval in which a fill or trade occurred.
75	TradeDate	Y	The system regards this as the logical business date for when the matched trade was submitted for clearing. Expressed in YYYYMMDD format.
60	TransactTime	Y	The UTC time when the trade was executed/created at the Exchange.

Tag	Field Name		Req'd	Comments
				If the TCR is sent regarding a trade bust, i.e. TradeReportType(856)=6, then TransactTime(60) will contain the UTC time when the trade was cancelled by RTC. UTC Date Time field. Format: YYYYMMDD-HH:MM:SS.sss
555	NoLegs		Y	Specifies the number of Instrument Legs (repeating groups) for a trade.
→	624	LegSide	Y	1 or 2 to correspond to Side (54).
→	20005	LegInstrument	Y	Corresponds to InstrumentLegNo.
→	20006	LegBuySell	Y	• B - Buy • S - Sell
→	5474	AbbreviatedPrice	N	Contract Price. Price of the leg can be expressed as: (b) A price code expressions (e.g. S + 10 which means settlement price plus ten) Valid price codes are B(Basis), S, C, M (mean), MC and M3 is also supported as tags and will be mapped to M if received. This is only valid for futures (c) A differential (e.g. -10 which means ten units lower than the price of the first leg) A valid differential is a number prefixed with either (+) or (-). Note, current installation does not handle Abbreviated prices for allowed instruments.
→	198	SecondaryOrderID	N	Can be used to provide order ID used by exchange or executing system. In RTC, the unique order ID provided by the trading system.
→	10003	LegLastQty	N	Details the trade quantity for this leg. This tag is introduced as ID 1418 from 5.0SP1.
→	637	LegLastPx	N	Represents the price for the current trade leg.
→	20007	UniqueTransaction ID	N	Unique Transaction Identifier, unique at half trade leg level.
552	NoSides		Y	Number of sides. The number of trade halves in the structure.
→	54	Side	Y	In RTC valid values: 1 = First side 2 = Second side
→	527	SecondaryExecID	N	A match ID assigned by RTC to each side of all matched trades. Different ID for buy and sell side.
→	37	OrderID	Y	OrderID is required to be unique for each chain of orders. In RTC, the unique client order id. Use “[N/A]” if not applicable.

Tag	Field Name		Req'd	Comments
→	11	ClOrdID	N	Client supplied order ID
→		**Parties**	N	For RTC, see 12.4
→	1	Account	Y	Account
→	581	AccountType	Y	Type of account associated with the trade. Valid Values: 1 = ISA 2 = House 3 = Omnibus 9 = Unallocated House Account (U-Account) 90 = Unallocated Client Account (X-Account)
→	5681	ExchangeTradeType	N	Exchange defined type of trade • 0 - NORMAL • 1 - BLOCK TRADE • 2 - EFRP • 3 - NIGHT TRADE
→	5476	PrivateReference	N	Private Reference up to 80 characters.
→	5477	PublicReference	N	Public Reference up to 12 characters.
→	5179	TradeTime	N	Time field representing the local time at the marketplace. Formats: • HH:MM:SS • HH:MM:SS.sss The value is copied from the TCR that was submitted to the FS.
→	58	Text	N	In RTC used to specify an optional reference on a trade.
	Standard Trailer		Y	

5.3.5 Trade Capture Report Variations

Below table show the differences in trade capture report field usage between the FIX servers and whether fields are required or not. TCR's from FSMD are described in a later section.

Tag		Name	FS		FSMEM	FSMD
			In	**Out**	**Out**	**Out**
		Standard Header	Y	Y	Y	Y
1180		ApplID	N	N	N	
1181		ApplSeqNum	N	N	N	
1350		ApplLastSeqNum	N	N	N	
571		TradeReportID	Y	Y	Y	Y

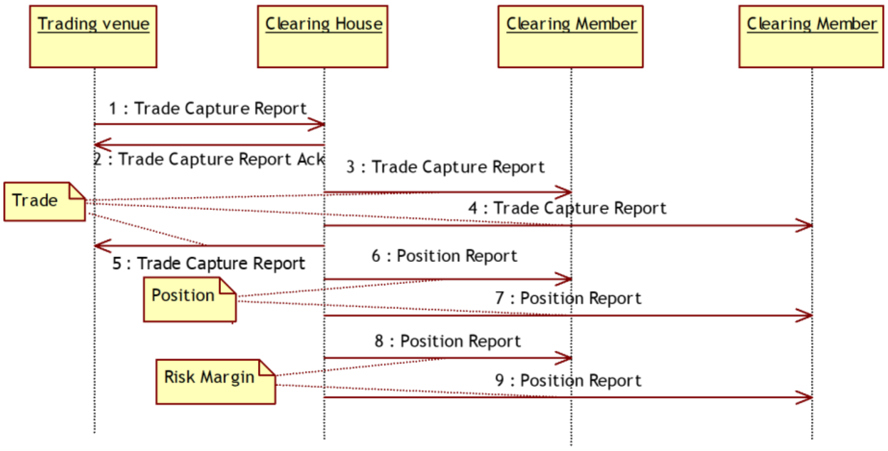
487		TradeReportTransType	Y	Y	Y	N
856		TradeReportType	Y	Y	Y	Y
568		TradeRequestID				N
855		SecondaryTrdType		N	N	N
150		ExecType		Y	Y	Y
748		TotNumTradeReports				N
912		LastRptRequested				N
325		UnsolicitedIndicator				N
572		TradeReportRefID	N	N	N	N
880		TrdMatchID		N	N	Y
17		ExecID	N	N	N	N
570		PreviouslyReported	Y	Y	Y	Y
5940		TradeNumber	N	N	N	N
7931		VenueID	N	N	N	N
1301		MarketID	Y	Y	N	N
943		TimeBracket	N	N	N	N
		Instrument	Y	Y	Y	Y
31		LastPx	N	N	N	N
75		TradeDate	Y	Y	Y	Y
60		TransactTime	Y	Y	Y	Y
555		NoLegs	Y	Y	Y	Y
→	624	LegSide	Y	Y	Y	Y
→	20005	LegInstrument	Y	Y	Y	Y
→	20006	LegBuySell	Y	Y	Y	Y
→	5474	AbbreviatedPrice	N	N	N	N
→	198	SecondaryOrderID	N	N	N	N
→	10003	LegLastQty	N	N	N	N
→	637	LegLastPx	N	N	N	N
→	5442	MatchingSlipID	N	N	N	N
→	20007	UniqueTransactionId	N	N	N	N
552		NoSides	Y	Y	Y	Y
→	54	Side	Y	Y	Y	Y
→	527	SecondaryExecID				N
→	37	OrderID	Y	Y	Y	Y
→	11	ClOrdID	N	N	N	N
→		**Parties**	N	N	N	N
→	1	Account	Y	Y	Y	Y
→	581	AccountType	Y	Y	Y	Y
→	5681	ExchangeTradeType	N	N	N	N
→	5476	PrivateReference	N	N	N	N
→	5477	PublicReference	N	N	N	N
→	5179	TradeTime	Y	Y	Y	Y
→	58	Text	N	N	N	N
→		**Standard Trailer**	Y	Y	Y	Y

5.4 Matched Trade Workflows

A matched trade may come from the trading system of a trading venue or a member. The trade report contains both sides of the trade and does not need to be acknowledged by the parties involved. (Please note that the outgoing trade reports, position reports, and risk margin reports are sent asynchronously and there is no guarantee in which order they will be delivered to the member).

5.4.1 Report Matched Trade from Trading Venue

This workflow describes reporting of a matched trade from a trading venue. This is a three-party trade report. The Trading venue reports a trade between two members.



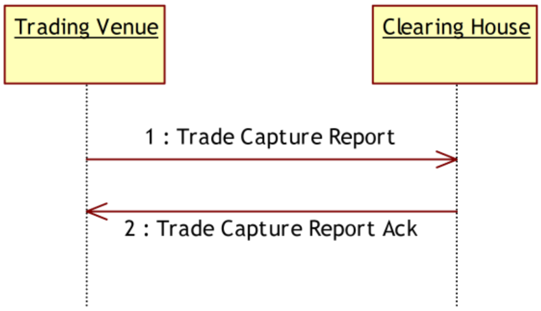
The system can reject message (1) with a Trade Capture Report Ack message.

The table below shows settings of specific fields in the TCR messages.

Msg	ExecType	TradeReportType	TradeReportTransType	TradeReportID	TradeReportRefID	TrdMatchID
-> 1 (TCR)	-	Submit	New	Client ID (a1)	-	-
<- 2 (TCR Ack)	New	Submit	New	Client ID (a1)	-	-
<- 3 (TCR) Buyer	Trade	Submit	New	RTC id	-	Trade ID in RTC
<- 4 (TCR) Seller	Trade	Submit	New	RTC id	-	Trade ID in RTC
<- 5 (TCR) Initiator	Trade	Submit	New	RTC id	Client ID (a1)	Trade ID in RTC

5.4.2 Reject Matched Trade

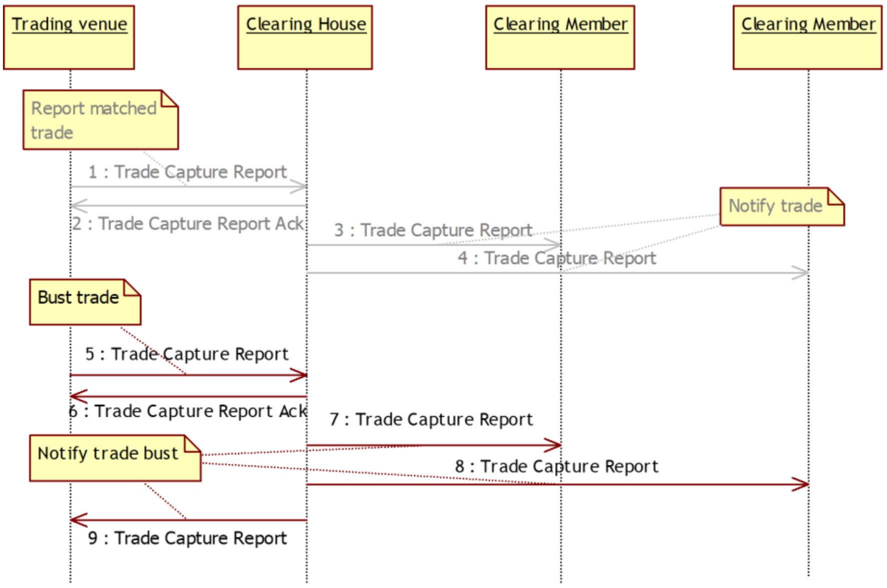
This workflow describes the rejection of a matched trade.



Msg	ExecType	TradeReportType	TradeReportTransType	TradeReportID	TrdRptStatus
-> 1 (TCR)	-	Submit	New	Client ID (a1)	
<- 2 (TCR Ack)	Rejected	Submit	New	Client ID (a1)	Rejected

5.4.3 Bust Matched Trade

The Bust Trade function is used for deleting trades that have been matched and captured by mistake. A busted trade may come from the trading system of a trading venue. The busted trade report has both sides of the trade and do not need to be acknowledged by the parties involved. The workflow starts at step 5. Step 1 to 4 is same as in section 5.4.1.

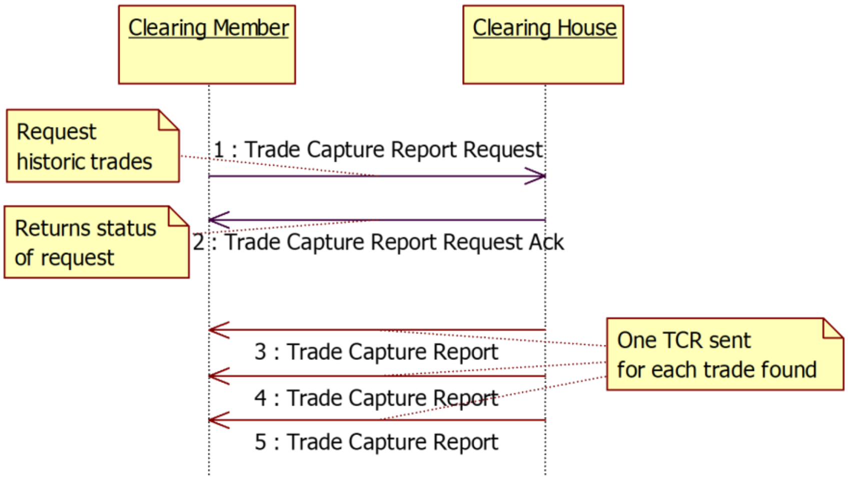


6 Trade Request and Subscription

This section describes the process of requesting trades and subscribing to new trades coming in from the exchange through the FSMD server.

6.1 Requesting Trades

Trades are requested by sending a TradeCaptureReportRequest(AD) message with filters specified as per the TradeCaptureReportRequest section. The message flow when requesting trades is illustrated in the picture below.


Clearing Member
Clearing House
Request historic trades
1 : Trade Capture Report Request
Returns status of request
2 : Trade Capture Report Request Ack
3 : Trade Capture Report
4 : Trade Capture Report
5 : Trade Capture Report
One TCR sent for each trade found

6.1.1 Requesting Historic vs Intraday Trades

The repeating group NoDates(580) is used to specify a date range when requesting historic trades. The absence of this group is interpreted as a request for intraday trades.

When requesting intraday trades, a snapshot of the current day’s active trades will be disseminated as one TCR per trade. Only TCRs for active trades will be sent, i.e. if a trade has been busted then no TCR will be sent for that trade.

When requesting historic trades, i.e. specifying the NoDates group, RTC will send the latest TCR for all trades that have been active during the date range. If a trade was submitted and busted within the specified date range the requester will receive the TCR cancellation message representing the bust.

6.2 Trade Subscription

The SubscriptionRequestType(263) is used for requesting subscriptions. The alternatives are:

- SubscriptionRequestType=0: Snapshot only (Default)
- SubscriptionRequestType=1: Snapshot and subscription
- SubscriptionRequestType=2: End subscription

When requesting a subscription, the subscriber will throughout the day receive TCRs for all trades matching the filter.

6.3 Business Message Types

RTC supports the message types described in the following table.

Fix Message name	Type	Dir.	Comment
Trade Capture Report	AE	In	Used for the following purposes: • Request cancellation of a matched trade (also known as trade break or bust)
Trade Capture Report	AE	Out	Used for the following purposes: • Sent unsolicited (when subscribed to) to indicate changes in trade state and trade event description • Sent as result of trade capture report request of historic trades
Trade Capture Report Ack	AR	Out	Used to acknowledge or reject an incoming Trade Capture Report
Trade Capture Report Request	AD	In	Used to request historic trades
Trade Capture Report Request Ack	AQ	Out	Used to acknowledge or reject a trade capture report request
Business Message Reject	j	Out	Used to reject malformed incoming requests/reports.

6.4 Message Details

See section 5.3.2 for Trade Capture Report Ack message specifications.

6.4.1 Trade Capture Report Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AD
568	TradeRequestID	Y	Identifier for the trade request
569	TradeRequestType	Y	Trade Request Type. Valid values 1 = Matched Trades
263	SubscriptionRequestType	N	Subscription Request Type. Valid values 0 = Snapshot(Default) 1 = Snapshot + Updates(Subscribe) 2 = Disable previous Snapshot + Update request (unsubscribe)
571	TradeReportID	N	To request a trade with a specific Trade Report ID.
17	ExecID	N	To request a trade with a specific Exchange assigned Execution ID (Trade Identifier)
818	SecondaryTradeReportID	N	To request a trade with a specific match identifier. A match identifier is assigned by RTC to each side of all matched trades. Different ID for buy and sell side. Use value from SecondaryExecID in TradeCaptureReport.
880	TrdMatchID	N	To request a trade with a specific match identifier. A match identifier is assigned by RTC to each side of all matched trades. Same ID for buy and sell side. Use value from the field with the same name in TradeCaptureReport.

Tag	Field Name	Req'd	Comments
	Parties		Used to specify the parties for the trades to be returned (clearing firm, execution broker, trader id, etc.) See section 12.4
	Instrument		See section 12.3
55	Symbol	N	The ID used for the tradable instrument. Can be used to filter down the requested positions.
711	NoUnderlyer	N	The number of Underlyingsymbols. Can be used to filter down the requested positions. Required if Underlyingsymbol is used
→ 311	UnderlyingSymbol	N	The underlying commodity. Can be used to filter down the requested positions. Required if NoUnderlying is used
580	NoDates	N	Number of date ranges provided. (RTC require 2 if specified to provide a range between two time periods) Absence of this repeating group indicates trades from the current trading day.
→	60 TransactTime	N	The UTC time when the trade was executed/created at the Exchange. Only the date portion of this filter is used. The time portion is ignored.
	Standard Trailer	Y	

6.4.2 Trade Capture Report Request Ack

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AQ
568	TradeRequestID	Y	Identifier for the trade request
569	TradeRequestType	Y	Trade Request Type 1 = Matched Trades
748	TotNumTradeReports	N	Number of trade reports matching search criteria
749	TradeRequestResult	Y	Result of Trade Request 0 = Successful 99 = Other
750	TradeRequestStatus	Y	Status of Trade Request 0 = Accepted 1 = Completed 2 = Rejected
	Instrument	N	See section 12.3
58	Text	N	Used to relay reason text if request is rejected.
	Standard Trailer	Y	

6.4.3 Trade Capture Report (Outgoing from RTC via FSMD)

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AE
571	TradeReportID	Y	Unique identifier for the Trade Capture Report.
487	TradeReportTransType	N	Identifies Trade Report message transaction type. Valid values: 0 = New
856	TradeReportType	Y	Type of Trade Report. Valid values: 0 = Submit 6 = Trade Report Cancel
568	TradeRequestID	N	Request ID if the Trade Capture Report is in response to a Trade Capture Report Request
855	SecondaryTrdType	N	Clearing trade event description.
150	ExecType	Y	Type of Execution being reported In RTC valid values: 8 = Rejected 9 = Suspended F = Trade H = Trade Cancel See section 5.2.6 for more information.
748	TotNumTradeReports	N	Number of trade reports returned - if this report is part of a response to a Trade Capture Report Request
912	LastRptRequested	N	Indicates if this is the last report in the response to a Trade Capture Report Request
325	UnsolicitedIndicator	N	Indicates whether or not message is being sent as a result of a subscription request
572	TradeReportRefID	N	The TradeReportID that is being referenced for some action, such as correction or cancelation.
880	TrdMatchID	Y	A match ID assigned by RTC to all matched trades. Same ID for buy and sell side.
17	ExecID	N	Exchange assigned Execution ID (Trade Identifier)
570	PreviouslyReported	Y	Indicates if the trade capture report was previously reported to the counterpart In RTC always set to N = not reported to counterpart
5940	TradeNumber	N	Where a trade is reported via Trading system the trade reference will be included here.
7931	VenueID	N	Venue of execution. Enumeration with value 0-4 supported.
	Instrument	Y	
31	LastPx	N	Option premium

Tag	Field Name		Req'd	Comments
943	TimeBracket		N	A code that represents a time interval in which a fill or trade occurred.
75	TradeDate		Y	The system regards this as the logical business date for when the matched trade was submitted for clearing. Expressed in YYYYMMDD format.
60	TransactTime		Y	The time when the trade was executed/created at the Exchange. UTC Date Time field. Format: YYYYMMDD-HH:MM:SS.sss
20009	TradeEventDescription		Y	Together with TradeReportType (tag 856) this field details the specific type of event. RTC valid values: 17 – New Trade Report 22 – Taken Up 23 – Given Up 56 – Busted 10009 – Trade Transfer In 10010 – Trade Transfer Out 10011 – Removed by Trade Split 10012 – Added by Trade Split 10013 – Partial Trade Transfer Out 10014 – Partial Trade Transfer In 10015 – Partial Give Up
555	NoLegs		Y	Specifies the number of Instrument Legs (repeating groups) for a trade.
→	624	LegSide	Y	1 or 2 to correspond to Side (54).
→	20005	LegInstrument	Y	Corresponds to InstrumentLegNo.
→	20006	LegBuySell	Y	• B – Buy • S – Sell
→	5474	AbbreviatedPrice	N	Contract Price. Price of the leg can be expressed as: (b) A price code expressions (e.g. S + 10 which means settlement price plus ten) Valid price codes are B(Basis), S, C, M (mean), MC and M3 is also supported as tags and will be mapped to M if received. This is only valid for futures (c) A differential (e.g. -10 which means ten units lower than the price of the first leg) A valid differential is a number prefixed with either (+) or (-). Note, current installation does not handle Abbreviated prices for allowed instruments.

Tag	Field Name		Req'd	Comments
→	198	SecondaryOrderID	N	Can be used to provide order ID used by exchange or executing system. In RTC, the unique order ID provided by the trading system.
→	10003	LegLastQty	No	Details the trade quantity for this leg. This tag is introduced as ID 1418 from 5.0SP1.
→	637	LegLastPx	No	The price of trade, supplied if the trade does not have field AbbreviatedPrice(5474) populated.
→	20007	UniqueTransactionID	N	Unique Transaction Identifier, unique at half trade leg level. Note, not used by RTC
552	NoSides		Y	Number of sides.
→	54	Side	Y	In RTC valid values: 1 = First 2 = Second
→	527	SecondaryExecID	N	A match ID assigned by RTC to each side of all matched trades. Different ID for buy and sell side.
→	37	OrderID	Y	OrderID is required to be unique for each chain of orders. In RTC, the unique client order id. Use “[N/A]” if not applicable.
→	11	ClOrdID	N	Client supplied order ID.
→		**Parties**		For RTC, see 12.4
→	1	Account	Y	Account
→	581	AccountType	Y	Type of account associated with the trade. Valid Values: 1 = ISA 2 = House 3 = Omnibus 9 = Unallocated House Account (U-Account) 90 = Unallocated Client Account (X-Account)
→	5681	ExchangeTradeType	N	Exchange defined type of trade • 0 - NORMAL • 1 - BLOCK TRADE • 2 - EFRP • 3 - NIGHT TRADE
→	5476	PrivateReference	N	Private Reference up to 80 characters.
→	5477	PublicReference	N	Public Reference up to 12 characters.
→	5179	TradeTime	Y	Time field representing the local time at the marketplace. Formats: • HH:MM:SS • HH:MM:SS.sss The value is copied from the TCR that was submitted to the FS.

Tag	Field Name		Req'd	Comments
→	58	Text	N	In RTC used as an optional trade reference.
	Standard Trailer		Y	

7 Position Management

7.1 Business Message Types

The RTC system supports the message types described in the following table.

Fix Message name	Type	Direction	Comment
Position Report	AP	Out	Used to: ▪ Relay the current position holdings
Position Maintenance Request	AL	In	Used to: ▪ Modify exercise quantity for an option.
Position Maintenance Report	AM	Out	Used to acknowledge or reject Position Maintenance Request
Request For Positions	AN	In	Used to request the current position holdings (and or subscribe to subsequent updates)
Request For Positions Ack	AO	Out	Used to acknowledge or reject an incoming Request For Positions

7.2 Position Account Content

If a subscription is active for the FIX session (via RequestForPosition), the content of each position account is disseminated in real-time. Each time the position is updated a position report is disseminated. It is also possible to query a snapshot of the current position holdings via the RequestForPositions message.

7.2.1 Main workflow

The Position Report message is used to disseminate the position account holding.

For each position the long position quantity and short position quantity is set.

A position account is indicated by setting the PosType=”TOT” in the PositionQty group.

7.3 Update Position Exercise Quantity

When an option expires the system calculates how much of a specific position is subject to automatic exercise. This volume can be adjusted by the member.

7.3.1 Main workflow

The Position Maintenance Request message is used to modify the option exercise quantity.

PosTransType is set to 1 (Exercise) and PosType is set to “EX” to indicate an update of the exercise quantity.

The LongQty field is used to specify the quantity eligible for exercise.

7.4 Message Details

7.4.1 Position Report

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AP
721	PosMaintRptID	Y	Unique identifier for this position report
710	PosReqID	N	Unique identifier for the Request for Positions associated with this report. This field should not be provided if the report was sent unsolicited.
325	UnsolicitedIndicator	N	Indicates whether or not a message is being sent as a result of a subscription request
728	PosReqResult	Y	Result of Request for Position RTC valid value: 0 = Valid Request
715	ClearingBusinessDate	Y	The Clearing Business Date
	Parties		See section 12.4
1	Account	N	Account. In RTC, conditionally required for positions. In RTC, the account field can indicate a Risk Calculation Node, a Risk Netting Group or an Ad-hoc Netting Group which are groups of accounts for which RTC calculates risk margin.
581	AccountType	N	Type of account associated with the trade. Valid Values: 1 = ISA 2 = House 3 = Omnibus 9 = Unallocated House Account (U-Account) 90 = Unallocated Client Account (X-Account)
	Instrument		
730	SettlPrice	N	Settlement price. In RTC, the MtM price. Not mandatory in RTC.
	PositionQty	Y	See section 12.5
	Standard Trailer	Y	

7.4.2 Position Maintenance Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AL
710	PosReqID	Y	Unique identifier for this position maintenance request as assigned by submitter.
709	PosTransType	Y	Identifies the type of the position transaction. In RTC valid values: 1 = Exercise
712	PosMaintAction	Y	1 = New
715	ClearingBusinessDate	Y	The Clearing Business Date referred to by this maintenance request
	Parties	Y	See section 12.4
1	Account	Y	Account.
581	AccountType	Y	
	Instrument	Y	

Tag	Field Name	Req'd	Comments
60	TransactTime	Y	Timestamp in UTC.
	PositionQty	Y	The position quantity applicable for the actions specified in PosTransType.
	Standard Trailer	Y	

7.4.3 Position Maintenance Report

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AM
721	PosMaintRptID	Y	Unique identifier for this position report
709	PosTransType	Y	Identifies the type of the position transaction. In RTC valid values: 1 = Exercise
710	PosReqID	N	Unique identifier for the position maintenance request associated with this report.
712	PosMaintAction	Y	1 = New
713	OrigPosReqRefID	Y	In RTC always set to “NONE”
722	PosMaintStatus	Y	0 = Accepted 2 = Rejected
715	ClearingBusinessDate	Y	The Clearing Business Date covered by this request
	Parties		See section 12.4
1	Account	Y	Account. In RTC, conditionally required for positions. In RTC, the account field can indicate a Risk Calculation Node, a Risk Netting Node or an Ad-hoc Netting Group which are groups of accounts for which RTC calculates risk margin.
581	AccountType	Y	Type of account associated with the order (Origin) In RTC, conditionally required for positions. RTC valid values: Type of account associated with an order Valid values: 1 = ISA 2 = House 3 = Omnibus 9 = Unallocated House Account (U-Account) 90 = Unallocated Client Account (X-Account)
	Instrument		
60	**TransactTime**	Y	The time this request was initiated/released by the trader, trading system or intermediary. The timestamp is in UTC.
	PositionQty	Y	The position quantity applicable for the actions specified in PosTransType.
58	Text	N	
	Standard Trailer	Y	

7.4.4 Request for Positions

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AN

Tag	Field Name	Req'd	Comments
710	PosReqID	Y	Unique identifier for the Request for Positions associated with this report.
724	PosReqType	Y	0 = Positions
263	SubscriptionRequestType	N	0 = Snapshot(default) 1 = Snapshot + Updates(Subscribe) 2 = Disable previous Snapshot + Update Request(Unsubscribe)
	Parties		See section 12.4
1	Account	N	Account.
581	AccountType	N	Type of account associated with the order (Origin) In RTC, conditionally required for positions. RTC valid values: Type of account associated with an order Valid values: 1 = ISA 2 = House 3 = Omnibus 9 = Unallocated House Account (U-Account) 90 = Unallocated Client Account (X-Account)
	Instrument		
715	ClearingBusinessDate	Y	Indicates the required clearing date for the request. Note that only the current day's positions are available via FIX. Expressed in YYYYMMDD format.
55	Symbol	N	The ID used for the tradable instrument. Can be used to filter down the requested positions.
711	NoUnderlyer	N	The number of Underlyingsymbols. Can be used to filter down the requested positions.
→ 311	UnderlyingSymbol	N	The underlying commodity. Can be used to filter down the requested positions. Required if NoUnderlyer(711) > 0.
60	TransactTime	Y	The UTC time when the trade was executed/created at the Exchange. Date Time field YYYYMMDD-HH:MM:SS.sss.
	Standard Trailer	Y	

7.4.5 Request for Positions Ack

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = AO
721	PosMaintRptID	Y	Unique identifier for this position report
728	PosReqResult	Y	0 - Valid Request 1 - Invalid or unsupported Request 3 - Not authorized to request positions 4 - Request for Position not supported 99 - Other

Tag	Field Name	Req'd	Comments
729	PosReqStatus	Y	0 - Completed 1 - Completed with Warnings 2 - Rejected
	Parties		See section 12.4
	Instrument		
	Standard Trailer	Y	

7.5 Workflows

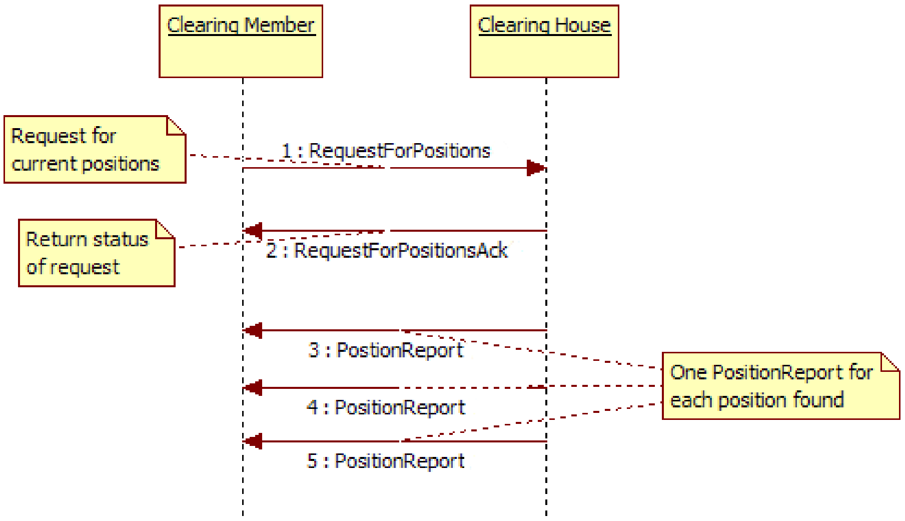
7.5.1 Update Position Exercise Quantity

When an option expires, the system calculates how much of a specific position that is subject to automatic exercise. This volume can be adjusted by the member. After a successful modification, a new Position Maintenance Report is disseminated with the requested exercise quantity.

Msg	PosMaintAction	PosTransType	PosType	PosTransType	PosMaintStatus	PosReqID	LongQty
-> 1 (PM Request)	0 = New	3 = Position adjustment	EX	3 = Position adjustment		Client ID (A1)	New quantity eligible for exercise
<- 2 (PM Report)	0 = New	3 = Position adjustment	EX	3 = Position adjustment	0 = Accepted	Client ID (A1)	New quantity eligible for exercise

7.5.2 Request for Positions

The Request for Positions function enables the user to receive current and subsequent updates of the position holdings by requesting a subscription via tag 263, SubscriptionRequestType set to 1(snapshot + updates). To only request current positions (i.e. not subsequent), SubscriptionRequestType shall be set to 0 (snapshot).



Msg	SubscriptionRequestType	PosReqType	PosReqResult	TotNumPosReports	PosReqID
-> 1 (Request FP)	1 = subscription	0 = Positions			Client ID (A1)
<- 2 (RFP Ack)			0 = Valid Request 1 = Invalid Request 99 = Other	Number of PRs that will be returned	Client ID (A1)
<- 3 (Position report)			0 = Valid Request		Client ID (A1)
<- 4 (Position Report)			0 = Valid Request		Client ID (A1)
<- 5 (Position Report)			0 = Valid Request		Client ID (A1)

8 Market Data In

This section describes how market data is fed to RTC via the FCMD server. The FCMD server connects as a client to an outside server and subscribes to market data.

If the external server is not online FCMD periodically tries to connect until the service is up.

The FCMD client uses the same password encryption as the other fix servers.

8.1 Market Data Message Types

The RTC system supports the message types described in the following table.

Fix Message name	Type	Direction	Comment
Market Data Request	V	Out	Used by the FCMD server to request a snapshot of market data.
Market Data Snapshot Incremental Refresh	X	In	Used by the market data service provider to send the requested market data.
Market Data Request Reject	Y	In	Used by the market data service provider to reject malformed Market Data Request messages.

8.1.1 Market Data Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = V
262	MDReqID	Y	Identifier for the market data request
263	SubscriptionRequestType	Y	Subscription Request Type 1 = Snapshot + Updates
264	MarketDepth	Y	Ignored, required by the FIX protocol
265	MDUpdateType	N	Market Data Update Type 1 = Incremental Refresh
267	NoMDEntryType	Y	Number of Market Data Entry Types in the request.
→ 269	MDEntryType	Y	The type of market data requested: ▪ 0 - Bid ▪ 1 - Offer ▪ 2 - Trade ▪ 6 - Settlement ▪ e - Evening closing ▪ r - Exchange rates
146	NoRelatedSym	N	Number of symbols requested
→ 55	Symbol	N	Symbol for requested instrument
	Standard Trailer	Y	

8.1.2 Market Data Snapshot Incremental Refresh

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = X
212	XmlDataLen	N	Part of the Standard Header. Length of the XmlData(213) field.
213	XmlData	N	Part of the Standard Header.
262	MDReqID	N	Identifier for the market data request, as set by the Market Data Request
268	NoMDEntries	Y	The number market data entries.
→ 279	MDUpdateAction	Y	Type of Market Data update action. ▪ 0 - New ▪ 1 - Change ▪ 2 – Delete
→ 269	MDEntryType	N	Specifies the entry type as supplied in the request
→	**Instrument**	N	Instrument block
→ 270	MDEntryPx	N	Price
→ 10013	MDEntryPxType	N	Valid values: 0 - Real-Time 1 - Provisional 2 – Confirmed 3 - Official 4 - Unofficial 5 - Daily 6 - Monthly Average 7 - Monthly Moving Average 8 - Indicative 9 - Reconfirmed
→ 15	Currency	N	Identifies currency used for price. Absence of this field is interpreted as the default for the security.
→ 271	MDEntrySize	N	Volume
→ 272	MDEntryDate	N	The date for the market data.
→ 273	MDEntryTime	N	The time for the market data
→ 277	TradeCondition	N	Space-delimited list of conditions describing a trade Valid values:

Tag	Field Name	Req’d	Comments
			U - Exchange Last (when update last paid) 0 - Cancel (Busted trade)
→ 10035	MDSource	N	Data source system
→ 574	MatchType	N	Valid value is: 4 - Auto-match
	Standard Trailer	Y	

8.1.3 Market Data Request Reject

Tag	Field Name	Req’d	Comments
	Standard Header	Y	MsgType = Y
262	MDReqID	Y	Identifier for the market data request, as set by the Market Data Request
281	MDReqRejReason	N	The reason why the request was rejected. 1 - Duplicate MDReqID 4 - Unsupported SubscriptionRequestType Y - Unknown Request Identifier Z - Other
58	Text	N	Free format text that optionally contains additional information regarding the rejection reason.
	Standard Trailer	Y	

8.2 Workflows

8.2.1 Request Exchange Rates

The request for exchange rates is done by using the MDEntryType 'r' and no NoRelatedSym group. The response is sent with no Instrument block and the currency in the Currency field.

9 Market Data Out

This section describes outgoing market data via FSMD. Market data published from FSMD is published in XML format within the FIX message.

9.1 Market Data Message Types

The RTC system supports the message types described in the following table.

Fix Message name	Type	Direction	Comment
Market Data Request	V	In	Used to request a snapshot of
Market Data Snapshot Full Refresh	W	Out	Used to send the requested market data.
Market Data Request Reject	Y	Out	Used to reject malformed Market Data Request messages.

9.2 Market Data Request

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = V
262	MDReqID	Y	Identifier for the market data request
263	SubscriptionRequestType	Y	Subscription Request Type 0 = Snapshot
264	MarketDepth	Y	Ignored, required by the protocol
265	MDUpdateType	N	Market Data Update Type 0 = Full Refresh
267	NoMDEntryType	Y	Number of Market Data Entry Types in the request, supported value is 1
→ 269	MDEntryType	Y	The type of market data requested: ▪ a - Commodity Settlement and Mean Prices (SMP) ▪ b - Futures Closing Prices (FCP) ▪ e - Traded Option Prices (TCP) ▪ g - Exchange Rates (EXR) ▪ h - Interest Rates (INR)
	Standard Trailer	Y	

9.3 Market Data Request Reject

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = Y
262	MDReqID	Y	Identifier for the market data request, as set by the Market Data Request
281	MDReqRejReason	N	The reason why the request was rejected. 1 - Duplicate MDReqID 4 - Unsupported SubscriptionRequestType Y - Unknown Request Identifier Z - Other

Tag	Field Name	Req'd	Comments
58	Text	N	Free format text that optionally contains additional information regarding the rejection reason.
	Standard Trailer	Y	

9.4 Market Data Snapshot Full Refresh

Tag	Field Name	Req'd	Comments
	Standard Header	Y	MsgType = W
212	XmlDataLen	N	Part of the Standard Header. Length of the XmlData(213) field.
213	XmlData	N	Part of the Standard Header.
262	MDReqID	N	Identifier for the market data request, as set by the Market Data Request
268	NoMDEntries	Y	The number of market data entries.
→ 269	MDEntryType	Y	Specifies the entry type as supplied in the request
→ 272	MDEntryDate	N	The date for the market data
	Standard Trailer	Y	

9.5 Workflows

9.5.1 Request market data

The market data request functionality enables the user to request all market data matching the criteria supplied in the request. The market data is supplied in XML format in the XmlData tag.

10 Recovery

10.1 Failure between RTC and Exchange

In the event of a system or communications failure between RTC and the exchange, the exchange will make available all the current day's unread messages until its normal end of the day.

The last 50,000 messages can be recovered using the standard FIX Resend Request message (section 2.5.4). To recover messages before that, the Application Message Request (see section 10.3) must be used.

There is no support for recovery within recovery.

10.2 Failure between RTC and Member or ISV

In the event of a system or communications failure between RTC and a member or ISV, RTC will make available the current day's state via snapshot for FSMD sessions or standard FIX Resend requests for FSMEM sessions.

If the failure persists beyond the normal RTC end of day, the member/ISV will be able to pick up trades on the following day by requesting the data for the previous business day's date. FIX does not support the availability of any other types of data using this method.

Trades, positions and other data can also be downloaded from the member/ISV's FTP directory.

10.3 Application sequencing and recovery

10.3.1 Application sequencing

The exchange system can run a set of parallel partitions and therefore each incoming TCR will include the partition ID in ApplID(1180) and the partition's sequence number in ApplSeqNum(1181).

The ApplSeqNum of each ApplID is reset to "1" at the start of each trading day.

Since only a subset of the TCRs processed per partition may be received, the field ApplLastSeqNum(1350) is also included in each TCR. This field will contain the ApplSeqNum of the last TCR generated. This can be used to distinguish deliberate sequence gaps from application errors by comparing the value of ApplLastSeqNum(1350) to the ApplSeqNum(1181) of the last received TCR from the same ApplID(1180).

10.3.2 Detecting an application sequence gap

Dropped messages can be detected by comparing the ApplLastSeqNum(1350) of each new TCR against the ApplSeqNum(1181) of the last TCR received from the same ApplID(1180).

10.3.3 Requesting retransmission of missed TCRs

The ApplicationMessageRequest(BW) can be used to recover any lost TRCs. The ApplReqType(1347) of the message should be 0 - Retransmission of Application Messages. The message may be used in on the following two modes:

- To request all TCRs after a particular TCR. The ApplSeqNum(1181) should be the application sequence number immediately after that of the last processed TCR.

- To request all TCRs for the day. The ApplSeqNum(1181) should be one (1).

In all cases, the field RefApplID(1355) should be used to identify the partition.

10.3.4 Request response

The exchange will respond to the Application Message Request with a Business Reject or a Session reject if the retransmission request is not successful with the reason specified.

In the case of a successful retransmission request, the exchange will resend the requested TCRs immediately.

10.3.5 Validation

A BusinessMessageReject message will be sent if an incorrect ApplID is specified, sequence number is out of range or application error at exchange side.

BusinessRejectReason(380)	Text(58)	Comment
0	404	ApplID does not exist
0	405	Application sequence number out of range
4		Application not available

10.4 Message Details

10.4.1 ApplicationMessageRequest

This message is used to recover missed messages when they are no longer available via the re-send request. RTC can detect missing messages by comparing the ApplLastSeqNum (1350) of each new TradeCaptureReport (AE) from the exchange against the ApplSeqNum (1181) of the previously received TradeCaptureReport (AE).

<table>
<tr><th>Tag</th><th colspan="2">Field Name</th><th>Req’d</th><th>Comment</th></tr>
<tr><td></td><td colspan="2"><b>Standard Header</b></td><td>Y</td><td>MsgType = BW</td></tr>
<tr><td>1346</td><td colspan="2">ApplReqID</td><td>Y</td><td>Client specified unique identifier of the request.</td></tr>
<tr><td>1347</td><td colspan="2">ApplReqType</td><td>Y</td><td>Type of request. Always set to 0.</td></tr>
<tr><td>1351</td><td colspan="2">NoApplIDs</td><td>Y</td><td>Number of ApplIDs to which the request relates.</td></tr>
<tr><td>→</td><td>1180</td><td>ApplID</td><td>N</td><td>Identifier of the partition.</td></tr>
<tr><td>→</td><td>1181</td><td>ApplSeqNum</td><td>N</td><td>Application sequence number of last message received sequence number.</td></tr>
<tr><td></td><td colspan="2"><b>Standard Trailer</b></td><td>Y</td><td></td></tr>
</table>

11 General Messages

Note, reports are not yet determined how it should be handled and what reports to be included. This section has therefore not been updated for that specific reason.

11.1 Business Message Types

Fix Message name	Type	Dir.	Comment
Business Message Reject	j	Out	Used to reject application messages which cannot be rejected by other means
News	B	Out	Used to relay news messages. Used for: - Market message - New static data - New market wide data - New member specific data - New exchange data - Margin call - New member report - End of trade - End of messages

11.2 Re-issued message messages and sequence number

When a user connects to the system all News messages relevant for the user since system start up will be re-issued. A News message have a sequence number set in Text(58) field so the user can distinguish between new and re-issued messages. Text(58) may contain more than just the sequence number, NoLinesOfText(33) field will indicated how many lines the Text(58) field will contain.

11.3 General Market message

A market Message can be sent by a clearing house user from cCran and will be sent unsolicited to all member FIX sessions.

- Urgency set to ‘Flash’.
- Headline will be set to ‘Broadcast’.
- A sequence number will be available in Text.
- Text set to ‘GEN’.
- Text will be set to the text of the market message.

11.4 New Static data

This message indicates that there is a change to part of the static dataset so the subscriber should collect and update static on the connecting system.

- Urgency set to ‘Flash’.
- Headline will be set to ‘New Static Data’.
- A sequence number will be available in Text.
- Product group, where applicable, will be available in Text.
  - Should specify “PG1 Data File”, “PG2 Data File”, or “Combined Data File”.

  - If the Product group is N/A, then the message “Combined Data File” should be added.

- Each file path generated will be available in Text. I.e. if there is both a csv and xml file generated, both filenames will be listed, one after the other.

Text(58) acronyms:

Text	Description	Product Group
CUR	Currencies	N/A
UND	Underlyings	PG1, PG2
CNT	Contract Types	PG1, PG2
SCT	Sub Contract Types	N/A
CON	Contracts	PG1, PG2
INS	Instruments	PG1, PG2
MEM	Member List	PG1, PG2, All
SCH	Schedule	N/A
MDT	Market Data Types	N/A
FRP	Fee Rules	N/A

11.5 New Market Wide data

This message indicates that there is a change to part of the market data so the subscriber should collect and update this on the connecting system.

- Urgency set to ‘Flash’.
- Headline will be set to ‘New Market Wide Data’.
- A sequence number will be available in Text.
- Product group, where applicable, will be available in Text.
  - Should specify “PG1 Data File”, “PG2 Data File”, or “Combined Data File”
  - If the Product group is N/A, then the message “Combined Data File” should be added.

- Each file path generated will be available in Text. I.e. if there is both a csv and xml file generated, both filenames will be listed, one after the other.

Text (58) acronyms:

Text	Description	Product Group
SPF	SPAN File	N/A
SET	Settlement Prices	PG1

SMP	Settlement and Mean Prices	PG1
FCP	Futures Closing Prices	PG1, PG2
ABP	Asian Benchmark Prices	N/A
NAP	Notional Average Prices	PG1
TCP	Traded Option Prices	PG1
CLO	Closing Prices	PG1, PG2, All
EXR	Exchange Rates	N/A
INR	Interest Rate	N/A
OPI	Open Interest	PG1, PG2
AEG	Auto Exercise Gradations	PG1, PG2

11.6 New Member data

This message indicates that there is member specific data available so the subscriber should collect and update this on the connecting system. This

- Urgency set to ‘Flash’.
- Headline will be set to ‘New Member Data’.
- A sequence number will be available in Text.

Text(58) acronyms:

- ‘TRD’ – Trades
- ‘OPP’ – Open Positions
- ‘IAS’ – Invoice and Account Sales
- ‘FDA’ – Financial Daily Activity
- ‘CLB’ – Non Cash Collateral Balances
- ‘CDA’ – Collateral Daily Activity
- ‘MDF’ – Member Default Fund
- ‘AEO’ – Auto Exercised Options
- ‘DCM’ – Delivery Commitments
- ‘EXP’ - Expiries
- ‘FEE’ – Fees (Daily)
- ‘ACC’ – Accounts
- ‘SMD’ – SPAN Margin Details
- ‘VMG’ - Variation Margins
- ‘AMG’ – Additional Margins
- ‘SMG’ – Summary Margins
- ‘FPA’ – Financial Pending Daily Activity
- ‘FSM’ – Financial Summary
- ‘COD’ – Cover Distribution
- ‘MFE’ - Fees (Monthly)
- ‘CFE’ – Collateral Fees (Daily)
- ‘MCF’ – Collateral Fees (Monthly)

- ‘CLS’ – Close Out Data File

11.7 New Exchange data

This message indicates that there is exchange specific data available so the subscriber should collect and update this on the connecting system. Only exchange users will receive this news message.

- Urgency set to ‘Flash’.
- Headline will be set to ‘New Exchange Data’.
- A sequence number will be available in Text.

Each file path generated will be available in Text. I.e. if there is both a csv and xml file generated, both filenames will be listed, one after the other.

Text (58) acronyms:

‘FEE’ – Fees

‘TRD’ – Trades

‘OPP’ – Open Positions‘ACC’ – Accounts

‘CFE’ – Collateral Fees

11.8 Margin Call Message

A Margin Call Message is indicated by sending a News message as follows:

- Urgency set to ‘Flash’.
- Headline set to ‘Margin Call’.
- A sequence number will be available in Text.
- Text set to ‘MRG’.
- The specific margin requirements in Text.

11.9 Report Availability Message

Report availability is indicated by sending a News message.

- Headline is set to ‘New Member Report’.
- A sequence number will be available in Text.

Text is set to the path and filename of the report that was generated. There will be one line per file generated (if multiple output formats have been selected in the report subscription, multiple files will be generated at once), the NoLinesOfText(33) field will indicated how many lines the Text(58) field will contain.

11.10 End of Trade

- Urgency set to ‘Flash’.
- **Headline** set to ‘End of Trade Messages’.
- A sequence number will be available in Text.
- Text set to ‘EOT’.

11.11 End of message

- Urgency set to ‘Flash’.
- Headline set to ‘End of Messages’.

- A sequence number will be available in Text.
- Text set to ‘EOM’.

11.12 Message Details

11.12.1 Business Message Reject

The Business Message Reject message can reject an application-level message which fulfils session-level rules and cannot be rejected via any other means, typically unsupported application messages or application messages lacking a specific reject message.

Tag	Field Name	Req’d	Comments
	Standard Header	Y	MsgType = j (lowercase)
45	RefSeqNum	N	*MsgSeqNum* of the rejected message.
372	RefMsgType	Y	*MsgType* of the rejected message.
379	BusinessRejectRefID	N	The value of the business level "ID" field on the message being referenced. Required unless the corresponding ID field (see list above) was not specified.
380	BusinessRejectReason	Y	0 = Other. 1 = Unknown ID. 2 = Unknown Security. 3 = Unsupported Message Type. 4 = Application not available. 5 = Conditionally Required Field Missing. 6 = Not authorized.
58	Text	N	Additional description of the message.
	Standard Trailer	Y	

11.12.2 News Message

Tag	Field Name	Req’d	Comments
	Standard Header	Y	MsgType = B
42	OrigTime	N	Time of message origination. Is in UTC.
61	Urgency	N	In RTC used values: 0 = Normal 1 = Flash (used as Alert in RTC)
148	Headline	Y	Specified the headline text
33	NoLinesOfText	Y	Identifies number of lines of text body
→	58 Text	Y	Additional description of the message.
	Standard Trailer	Y	

12 Common Component Blocks

All Session and Application Messages use the Standard Message Header and Trailer, as specified in the FIX 4.4 Specification. The following sections describe the FIX fields that are supported by the RTC.

12.1 Standard Header

Tag ID	Name	Req'd	Comments
8	BeginString	Y	FIX.4.4 (must be first field in message)
9	BodyLength	Y	(must be second field in message)
35	MsgType	Y	(must be third field in message)
49	SenderCompID	Y	
56	TargetCompID	Y	
34	MsgSeqNum	Y	
43	PossDupFlag	N	Always required for retransmitted messages, whether prompted by the sending system or as the result of a resend request.
52	SendingTime	Y	Timestamp is in UTC. Note, should match system date/time.
122	OrigSendingTime	N	Required for message resent as a result of a ResendRequest. If data is not available set to same value as SendingTime. Timestamp is in UTC.
369	LastMsgSeqNumProcessed	N	The last message sequence number received (based on Tag 34).

12.2 Standard Trailer

Tag ID	Name	Req'd	Comments
10	CheckSum	Yes	A three byte simple checksum. See FIX 4.4 Specification for a detailed description. Always the last field in the message.

12.3 Instrument

The Symbol field represents a unique tradable instrument.

Tag	Field Name		Req'd	Comments
55	Symbol		***	The ID used for the tradable instrument.
10010	NoOfInstrumentLegs		Y	The number of instrument legs contained within the message. 1 unless a multi-legged carry trade.
→	20004	InstrumentLegNo	Y	1 to n.

→	5475	ExpiryDate	N	Expiry Date for futures contracts in DDMMYY format. For Options contracts, this field will be populated by the expiry month code in MMMYY format.
→	202	StrikePrice	N	Used for derivatives, such as options
→	20008	UniqueProductID	N	Unique Product Identifier
461	CFICode		Y	Indicates the type of security using ISO 10962 standard, Classification of Financial Instruments (CFI code) values.

12.4 Parties

The parties block identifies the parties involved in a business transaction. For use in RTC, refer to the description of the party role below and the mapping to the RTC parties, as well as the description in 5.2.5.

Tag	Field Name		Req 'd	Comments
453	NoPartyIDs		N	Number of party id's
→	448	PartyID	N	

→	447	PartyIDSource	N	In RTC set to:
				D = Proprietary/Custom code
				N = Legal Entity Identifier

→	452	PartyRole	N	

The valid PartyRoles are as follows:

Tag	Field Name	Data type	Description
452	PartyRole	int	Identifies the type or role of the PartyID (448) specified. Valid values: 3 = Client ID 4 = Clearing Firm (RTC: Clearing Member) 24 = Customer Account (RTC: Client LEI) 36 = Entering trader (RTC: Trading User) 76 = Desk ID (RTC: Business Unit)

12.5 PositionQty

Tag	Field Name		Req'd	Comments
702	NoPositions			Number of positions.
→	703	PosType	N	Required if NoPositions > 0. In RTC, the valid values:

				TOT = Total Transaction Qty. In RTC the Settlement Position. Used when the M-t-M is defined as ‘Firm’. EX = Option Exercise Qty. In RTC used to indicate the quantity eligible for option exercise.
→	704	LongQty	N	
→	705	ShortQty	N	
→	706	PosQtyStatus	N	

13 User Defined Tags

For RTC, the FIX Protocol is extended with the following user defined tags:

Tag	Field Name	Comments
20000	AccessGroupId	Access group ID. In RTC, an Account belongs to exactly one Access Group.
20001	RiskNettingGroupId	Risk Netting Group ID. In RTC, an Account belongs to exactly one Risk Netting Group.
20002	RiskCalculationNodeId	Risk Calculation Node ID. In RTC, an Account belongs to exactly one Risk Calculation Node but one RCN may hold many accounts.
20003	AdHocNettingGroupId	Ad-hoc Netting Group ID. In RTC, an Account may belong to multiple Ad-hoc Netting Groups.
20004	InstrumentLegNo	As NoLegs
20005	LegInstrument	Corresponds to InstrumentLegNo.
20006	LegBuySell	B - Buy, S - Sell.
20007	UniqueTransactionID	Unique Transaction Identifier, unique at half trade leg level.
20008	UniqueProductID	Unique Product Identifier.
20009	TradeEventDescription	Detailed description of trade event. Should be used together with TradeReportType (tag 856). RTC valid values: 17 – New Trade Report 22 - Taken Up 23 - Given Up 56 - Busted 10009 - Trade Transfer In 10010 - Trade Transfer Out 10011 - Removed by Trade Split 10012 - Added by Trade Split 10013 - Partial Trade Transfer Out 10014 - Partial Trade Transfer In 10015 - Partial Give Up